

AMENDMENTS TO THE CLAIMS

Claims 1-59 (Canceled)

60. (Original) A method for operating an operational data store, comprising:
creating a new partition in a composite-partitioned history table;
creating a partitioned temporary table;
filling the temporary table with data from an insert table;
exchanging the temporary table with the new partition; and
receiving a query and applying the query to both the history table and the insert table.
61. (Original) The method of claim 60, further comprising:
creating a new partition in the insert table based on values from an existing partition; and
dropping the existing partition.

Claims 62-68 (Canceled)

69. (Original) A computer program product for operating an operational data store, the computer program product comprising a computer usable medium having computer readable code thereon, including program code which:
creates a new partition in a composite-partitioned history table;
creates a partitioned temporary table;
fills the temporary table with data from an insert table;
exchanges the temporary table with the new partition; and
receives queries and applies said queries to both the history table and the insert table.
70. (Original) A system for producing a desired level of service in a mixed workload environment, comprising:

a high-speed insert operational data store (ODS);
a throttler for throttling selected transactions to the ODS; and
an aggregator for accumulating transactions into batches and inserting each of the batches into the ODS using a single database transaction per batch.

71. (Original) The system of Claim 70, wherein the mixed workload environment includes at least two of archiving, OLTP queries, DSS queries, high-speed inserts, backup processes and extract/translate/load transactions.

72. (Original) A method for producing a desired level of service in a mixed workload environment, comprising:

insert transactions into an operational data store (ODS) at a high-speed;
throttling selected transactions to the ODS;
accumulating transactions into batches; and
inserting each of the batches into the ODS using a single database transaction per batch.

73. (Original) The method of Claim 72, wherein the mixed workload environment includes at least two of archiving, OLTP queries, DSS queries, high-speed inserts, backup processes and extract/translate/load transactions.

74. (New) An operational data store, comprising:

an insert table for storing new data;
a history table for storing historical data, said history table comprises:
a number of partitions, wherein each of said number of partitions is partitioned into a number of sub-partitions equal to the number of database server instances, and wherein said history table is partitioned into at least one of said number of partitions by a range; and
transfer logic for periodically transferring new data from the insert table to the history table.

75. (New) The operational data store of Claim 74, wherein each of said number of sub-partitions of each of said number of partitions is associated with said database server instance.
76. (New) The operational data store of Claim 74, the transfer logic comprising:
a secondary table;
fill logic for filling the secondary table with selected data from the insert table; and
secondary transfer logic for transferring the secondary table into the history table, the selected data thereby being transferred into the history table.
77. (New) The operational data store of Claim 76, wherein the history table has an indexing scheme, the secondary transfer logic further comprising:
indexing logic for applying the history table indexing scheme to the secondary table, wherein the indexing logic applies the history table indexing scheme to the secondary table prior to transferring the secondary table into the history table.
78. (New) The operational data store of Claim 76, the secondary transfer logic further comprising:
table logic for creating a new partition the history table, the new partition for swapping with the secondary table, wherein the secondary transfer logic swaps the secondary table and the new partition by exchanging respective pointers.
79. (New) The operational data store of Claim 74, further comprising:
a query engine for applying a database query to both the history table and the insert table.
80. (New) The operational data store of Claim 74, further comprising:
an aggregation buffer for accumulating new data; and
an aggregator for batching the accumulated data and transferring the batched data into the insert table with a single database access.

81. (New) The operational data store of Claim 80, wherein the aggregator transfers a batch of new data into the insert table when a batch size surpasses a maximum size, wherein the batch size is configurable.

82. (New) The operational data store of Claim 81, wherein the batch size is measured according a group consisting of: a number data bytes and a number of records.

83. (New) The operational data store of Claim 80, wherein the aggregator transfers batches of new data into the insert table at regular intervals, defined by a given period, wherein said period is configurable.

84. (New) The operational data store of Claim 80, wherein the aggregator transfers batches of new data into the insert table when the aggregation buffer surpasses a given maximum buffer size, wherein the maximum buffer size is configurable.

85. (New) An operational data store, comprising:
an insert table for storing new data;
a history table for storing historical data;
transfer logic for periodically transferring new data from the insert table to the history table; and
a throttler for throttling transactions of different classes independently to achieve a desired level of service.

86. (New) The operational data store of Claim 85, wherein a first transaction class is a query, and a second transaction class is an insert, the throttler throttling queries so that inserts can be executed at at least the desired level of service.

87. (New) The operational data store of Claim 85, wherein at least one transaction class comprises plural transaction types, the throttler throttling transactions of different types independently.

88. (New) The operational data store of Claim 85, further comprising:
a plurality of processor nodes configured as a processor cluster, wherein distinct database server instances are associated with distinct processor nodes of the processor cluster.
89. (New) The operational data store of Claim 88, wherein the history table is partitioned, each partition is further sub-partitioned into a number of sub-partitions, and the number of sub-partitions is equal to the number of database server instances.
90. (New) The operational data store of Claim 85, wherein data from the insert table is transferred to the history table at regular intervals, wherein the intervals are selected from the group consisting of: configurable intervals and different intervals for different tables.
91. (New) A method for maintaining an operational data store, comprising:
inserting new data into an insert table;
partitioning the history table into a number of partitions according to a range;
partitioning each of said number of partitions into a number of sub-partitions equal to a number of database server instances, and
periodically transferring data from the insert table to a history table.
92. (New) The method of Claim 91, further comprising:
associating each said number of sub-partitions of each said number of said partitions with the database server instance.
93. (New) The method of Claim 91, further comprising:
creating a secondary table;
filling the secondary table with selected data from the insert table; and
transferring the secondary table into the history table, the selected data thereby being transferred into the history table.

94. (New) The method of Claim 93, wherein the history table has an indexing scheme, the method further comprising:
applying the history table indexing scheme to the secondary table, wherein the history table indexing scheme is applied to the secondary table prior to transferring the secondary table into the history table.
95. (New) The method of Claim 93, further comprising:
creating a new partition in the history table, wherein the secondary table is transferred by being swapped with the new partition, wherein the secondary table and new partition are swapped by exchanging respective pointers.
96. (New) The method of Claim 91, further comprising:
applying a database query to both the history table and the insert table.
97. (New) The method of Claim 91, further comprising:
aggregating new data into batches; and
inserting the batched new data into the insert table with a single database access.
98. (New) The method of Claim 97, wherein a batch of new data is transferred into the insert table when a batch surpasses a maximum size, wherein the maximum size is configurable.
99. (New) The method of Claim 98, wherein batch size is measured according to a group consisting of: a number data bytes and a number of records.
100. (New) The method of Claim 97, wherein batches of new data are transferred into the insert table at regular intervals, defined by a given configurable period.
101. (New) The method of Claim 97, further comprising:

aggregating the batches of new data in an aggregation buffer, wherein the batches are transferred into the insert table when the aggregation buffer surpasses a given configurable maximum buffer size.

102. (New) A method for maintaining an operational data store, comprising:
inserting new data into an insert table;
periodically transferring data from the insert table to a history table; and
throttling transactions of different classes independently to achieve a desired level of service.
103. (New) The method of Claim 102, wherein a first transaction query is a query, and a second transaction class is an insert, and queries are throttled so that new data can be inserted at at least the desired level of service.
104. (New) The method of Claim 102, wherein at least one transaction class comprises plural transaction types which are independently throttled.
105. (New) The method of Claim 102, further comprising:
configuring plural processor nodes as a processor cluster; and
executing distinct database server instances on distinct processor nodes of the processor cluster.
106. (New) The method of Claim 105, further comprising:
partitioning the history table; and
sub-partitioning each partition into a number of sub-partitions, wherein the number of sub-partitions is equal to the number of database server instances.
107. (New) An operational data store, comprising:
means for inserting new data into an insert table;
means for periodically transferring data from the insert table to a history table;

means for applying a database query to both the history table and the insert table; and

means for throttling transactions of different types independently to achieve a desired level of service.

108. (New) The operational data store of Claim 107, further comprising:

means for batching new data; and

means for inserting the batched new data into the insert table with a single database access.

109. (New) An operational data store, comprising:

means for creating a new partition in a composite-partitioned history table;

means for creating a partitioned temporary table;

means for filling the temporary table with data from an insert table;

means for exchanging the temporary table with the new partition;

means for receiving a database query and applying said query to both the history table and the insert table; and

means for throttling transactions of different types independently to achieve a desired level of service.

110. (New) A computer program product for operating an operational data store, the computer program product comprising a computer usable medium having computing reading code thereon, including program code which:

inserts new data into an insert table;

periodically transfers data from the insert table to a history table;

applies a database query to both the history table and the insert table; and

throttles transactions of different types independently to achieve a desired level of service.

111. (New) The computer program product of Claim 110, wherein the program code further:

batches new data; and

inserts the batched new data into the insert table with a single database access.